EFFICIENT ANALYTICAL INTEGRATION OF SINGLE SCATTERING FUNCTION

Umashankar Pradhan and Subodh Kumar
Dept. of Computer Sc. & Engg., IIT Delhi, New Delhi, India

Keywords: Global Illumination, Scattering, Participative Media, Real time rendering.

Abstract: Light scattering through a participative medium has a significant impact on display. However, accurate and efficient simulation of scattering remains challenging. Monte Carlo or numerical integration techniques are commonly employed to solve the scattering equation. Single scattering is a common approximation that provides satisfactory results in many cases. Analytical integration of scattering under certain assumptions have been achieved by pre-computing a table of values. We present a new approximation to the single scattering equation that is easily integrable in real time. We analyze the error of this approximation and show that the numerical error is insignificant. The images are virtually indistinguishable from those obtained by the more accurate integration.

1 INTRODUCTION

Models of scattering are employed in computer graphics for accurate display of phenomena from fog, smoke, snow and cloud to hair, marble and skin. Optical scattering refers to deviation of light from the straight path due to particles in the medium as well as to deviation from the specular direction after reflection. Scattering is important for realistic display. Indeed, most uncontrolled environments exhibit sufficient scattering to induce glows around light source or to somewhat blur appearance of objects. Light shafts and haziness, for example, are effects of scattering common in photographs but often missing from computer generated imagery because they are complex and slow to compute.

In this paper we focus on light scattering when traveling through participative media. We assume particles are significantly larger than light's wavelength, focussing on geometric scattering. Usually light scatters multiple times as it encounters particles in its path. However, often single scattering approximations are used, which accurately describe sparse media – but are often useful for dense media as well.

Generic single scattering is described in Figure 1. Light from source *l* is scattered by a particle at *p*. A fraction scatters towards viewpoint *e*. Such scattering takes place from all points along any direction of view *v*. We integrate these to obtain the total scattered radiance. This is often referred to as the airlight integral. The total radiance also includes any source (including reflective surface) in the direction of view. Some of this also scatters away and does not reach the viewpoint. The radiance scattering straight ahead is separately integrated.

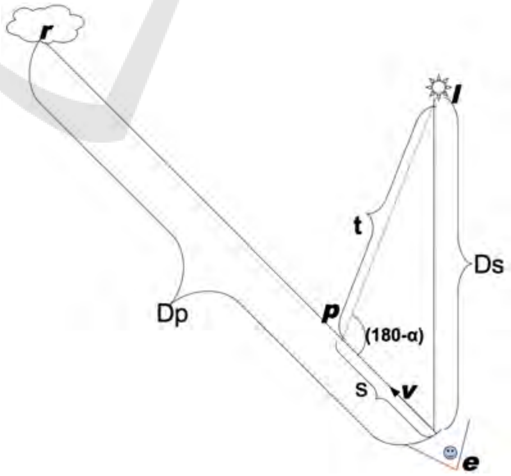

Figure 1: Standard formulation of single scattring.

Integration of the airlight function can be performed by numerical techniques. However, when analytical integration is possible, images can be generated significantly faster. Ours is an attempt belonging to this genre. We approximate the scattering function by analytically integrable ones and show that the error is small, quantitatively as well as qualitatively.

Pradhan U. and Kumar S..
EFFICIENT ANALYTICAL INTEGRATION OF SINGLE SCATTERING FUNCTION.
DOI: 10.5220/0003373901930199
In *Proceedings of the International Conference on Computer Graphics Theory and Applications* (GRAPP-2011), pages 193-199
ISBN: 978-989-8425-45-4

2 RELATED WORK

Importance of scattering has been recognized long in computer graphics (Blinn, 1982). Even analytical models were described early (Max, 1986; Nishita et al., 1987). Light transport equations (Chandrasekhar, 1960) are often solved using Monte Carlo techniques (Stam, 1994; Lafortune, 1996; Jensen and Christensen, 1998) and finite element method (Rushmeier and Torrance, 1987; Sillion, 1994; Arbree et al., 2010a; Arbree et al., 2010b). Both are expensive.

(Premoze et al., 2004) use point spread functions to compute airlight due to multiple scattering. (Sun and Ramamoorthi, 2005) attempt to perform the airlight integral analytically and our work is inspired by theirs. The integral however does not admit analytical solution. (Sun and Ramamoorthi, 2005) hence precompute a 2D table, assuming that the medium density and scattering coefficients are known. The intermediate values are then evaluated using interpolation. We, instead, derive functions that track the integral for different values of medium density and particle sizes. And, we do not require precomputation at a pre-determined resolution. Further, we incorporate backward scattering as well. Our work is also similar in spirit to that of (Biri et al., 2006) who use a polynomials of degree four to approximate the kernel. However, the function is not well approximated by polynomials and their approximation incurs unacceptably large error (as shown later).

Later (Zhou et al., 2007) provide an analytic approximation to the airlight integral in the presence of inhomogeneous media whose density can be described as a sum of Gaussians. The airlight integral is performed for each Gaussian. More recently (Bernabei et al., 2010) propose a spherical harmonics based representation of inhomogeneous material properties. They subdivide inhomogenous media into voxels and sample for each voxel, the attenuation of light in various directions. Spherical harmonics based representation of these samples is used to later integrate along a ray. In the next sections we describe our approximation and present error analysis and rendering results.

3 SINGLE SCATTERING INTEGRAL

This table lists the terminology for reference (see also Figure 1):

I_s =	Radiant intensity of the point light source
D_s =	Distance of the light source from the integration point (view point)
θ_s =	Angle between the direction of integration and the direction to light
α =	Phase angle of scattering
$P(\alpha)$ =	Phase function for anisotropic scattering
x =	Distance of the scattering point from the perpendicular to light source (forward scattering)
t_x =	Distance of the light source from the point of forward scattering
y =	Distance of the scattering point from the perpendicular to light source (backward scattering)
t_y =	Distance of the light source from the point of backward scattering

The radiance L comprises the direct transmission component L_d and the single scattering radiance (airlight), L_a: $L = L_d + L_a$. The direct term L_d simply attenuates the incident radiance from a point source by an exponential corresponding to the distance between the source and the viewer and the scattering coefficient β:

$$L_d(\theta_s, D_s, \beta) = \frac{I_s}{D_s^2} e^{-\beta D_s} \delta(\theta_s),$$

where the impulse function δ indicates that the direct component is non-zero only in the direction of the light (i.e., point light source or reflective point on a surface). The airlight component (Nishita et al., 1987) is:

$$L_a(\theta_s, D_s, D_p, \beta) = \int_0^{D_p} \beta P(\alpha) \frac{I_s e^{-\beta t}}{t^2} e^{-\beta s} ds,$$

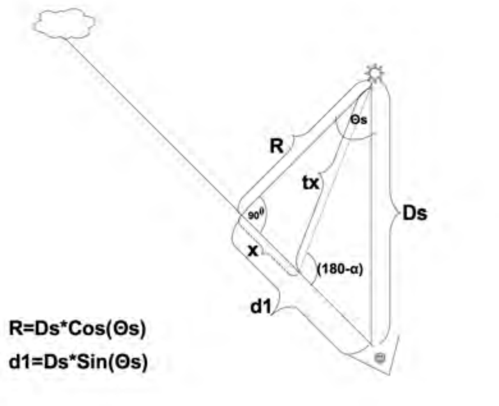

Figure 2: Forward Scattering.

where D_p is the distance to the closest surface point along the view direction, or ∞ if no such surface exists. The integration is over single scattering at distance s from the viewer and t is the distance of the scattering point from the source of light. Subdividing the integral into parts L_1 (forward scattering) and L_2 (backward scattering):

$$L_1(\theta_s, D_s, D_p, \beta) = \int_0^{d_1} \beta P(\alpha) \frac{I_s e^{-\beta t_x}}{t_x^2} e^{-\beta(d_1 - x)} dx,$$

where $d_1 = D_s \sin\theta_s$. Figure 2 illustrates the case when the scattering angle α is less than 90° and Figure 3 does when the angle is more than 90°.

$$L_2(\theta_s, D_s, D_p, \beta) = \int_0^{D_p} \beta P(\alpha) \frac{I_s e^{-\beta t_y}}{t_y^2} e^{-\beta(y + d_1)} dy.$$

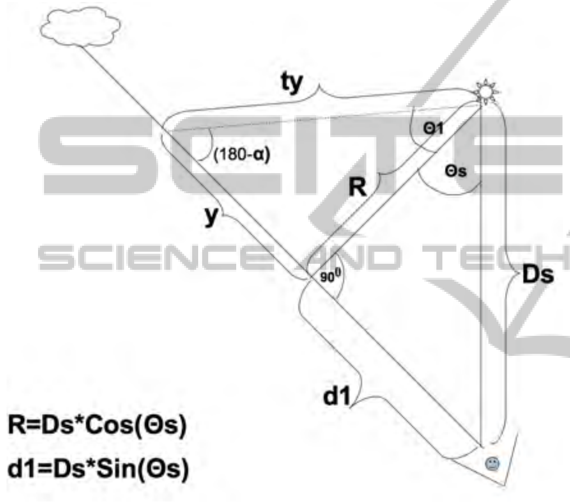

Figure 3: Backward Scattering.

Consider L_1 first, with the default isotropic phase function:

$$L_1(\theta_s, D_s, D_p, \beta) = \frac{\beta I_s}{4\pi} \int_0^{d_1} \frac{e^{-\beta t_x}}{t_x^2} e^{-\beta(d_1 - x)} dx,$$

with $t_x^2 = x^2 + R^2$ (and $R = D_s \cos\theta_s$). Thus:

$$L_1(\theta_s, D_s, D_p, \beta) = \frac{\beta I_s}{4\pi} \int_0^{d_1} \frac{e^{-\beta(\sqrt{x^2+R^2} + d_1 - x)}}{t_x^2} dx$$

With $x = R\tan\theta$:

$$L_1(\theta_s, D_s, D_p, \beta) = \frac{\beta I_s}{4\pi R} \int_0^{\theta_s} e^{-\beta R(\sec\theta - \tan\theta)} d\theta$$

Similarly,

$$L_2(\theta_s, D_s, D_p, \beta) = \frac{I_s \beta}{4\pi} \int_0^{D_p} \frac{I_s e^{-\beta t_y}}{t_y^2} e^{-\beta(y + d_1)} dy. \quad (1)$$

And with $t_y^2 = R^2 + y^2$ and $y = R\tan\theta$,

$$L_2(\theta_s, D_s, D_p, \beta) = \frac{I_s \beta}{4\pi R} e^{-\beta d_1} \int_0^{\theta_1} e^{-\beta R(\sec\theta + \tan\theta)} d\theta, \quad (2)$$

where θ_1 is the angle made by the vector from the light to surface and the perpendicular to the direction of view as shown in Figure 3. If there is no surface, θ_1 is $\frac{\pi}{2}$.

3.1 Forward Scattering Approximation

In order to compute the integrals in equations 1 and 2, we use the following approximation to $\sec\theta - \tan\theta \approx$

$$\begin{array}{ll} -0.8405\theta + 0.9915, & 0 \le \theta \le 0.3959 \\ -0.6331\theta + 0.9094, & 0.3959 \le \theta \le 0.8395 \\ -0.5216\theta + 0.8158, & 0.8395 \le \theta \le \frac{\pi}{2} \end{array}$$

This is a low error piecewise linear approximation to $\sec\theta - \tan\theta$. Now this function can by analytically integrated. The error is less than the precomputed table of (Sun and Ramamoorthi, 2005) once the entire integral is evaluated. Figure 4 shows the relative error in our approximation. As a comparison, the polynomial based error (Biri et al., 2006) is many times higher as shown in Figure 5

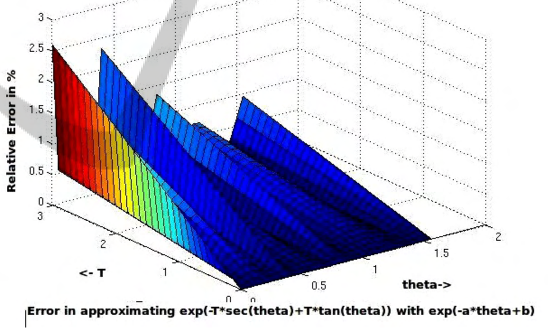

Figure 4: Relative error in overall radiance due to the $\sec\theta - \tan\theta$ approximation.(T = βR).

Although isotropic scattering phase function is commonly used in many implementations, the Henyey Greenstein phase function approximation (Henyey and Greenstein, 1941):

$$P(\alpha) = \frac{1 - g^2}{(1 + g^2 - 2g\cos\alpha)^{\frac{3}{2}}}$$

produces more realistic results. Here g is usually taken as a constant property of the medium. This can be incorporated into our system again by using a Gaussian like form: $P(x) \approx ae^{-bx}$ where a and b are quadratic functions of g:

$$a = a_1 g^2 + a_2 g + a_3$$

$$b = b_1 g^2 + b_2 g + b_3$$

Now a_i and b_i are chosen to provide the best fit, still ensuring a small error. For the error to remain

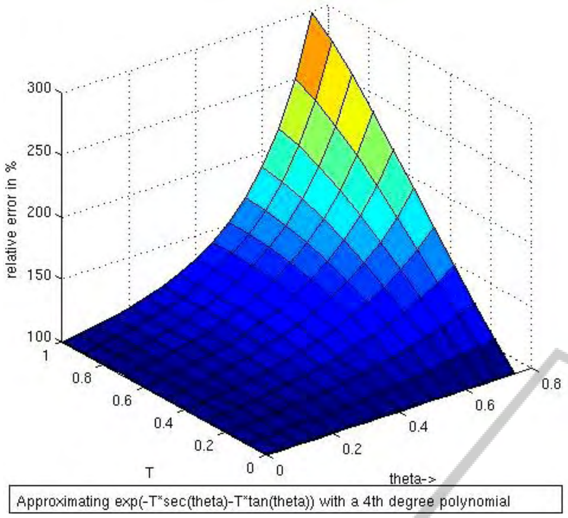

Figure 5: Relative error in radiance due to polynomial approximation (Biri et al., 2006).(compare to Figure 4.)

bounded, different a_i and b_i values are needed for different ranges of g. The detailed set of coefficients, which generate low error, are reproduced in the appendix. Note that the value of g will be constant for a uniform medium and known at the rendering time and hence only one of these will be used. We show here the error for a few values of g. Recall that g lies between 0 and 1, g near 0 for little scattering and g near 1 for denser media like haze and fog. The maximal error we observe is for lower values of g and it reduces as g increases.

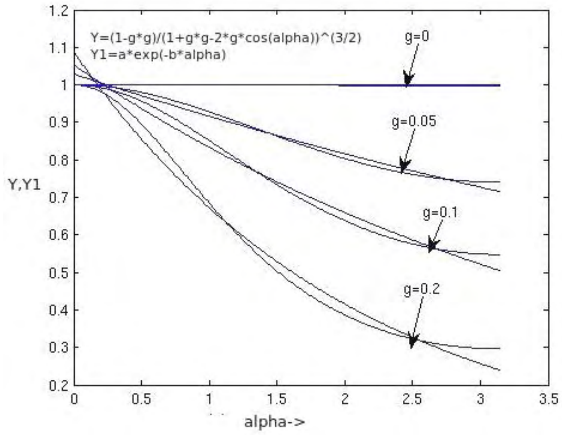

Figure 6: Comparing our approximation to the Henyeye Greenstein approximation for increasing values of phase angle α: $0 \leq g \leq 0.2$. The original and our approximation curves are shown. The maximum error is for this range of g and occurs near $\alpha = 0$. Values of g less than 0.5 are not commonly used.

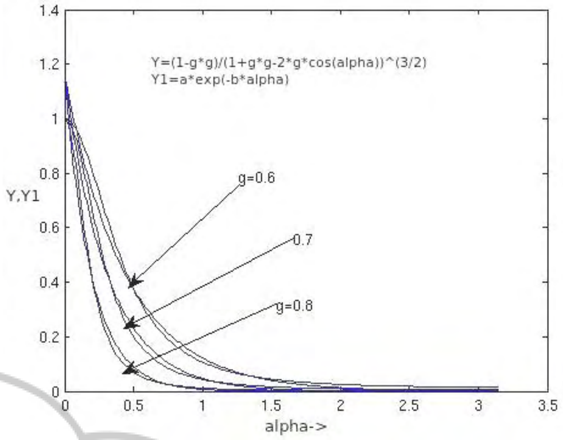

Figure 7: Comparing our approximation to the Henyeye Greenstein approximation for increasing values of phase angle α: $0.6 \leq g \leq 0.8$.

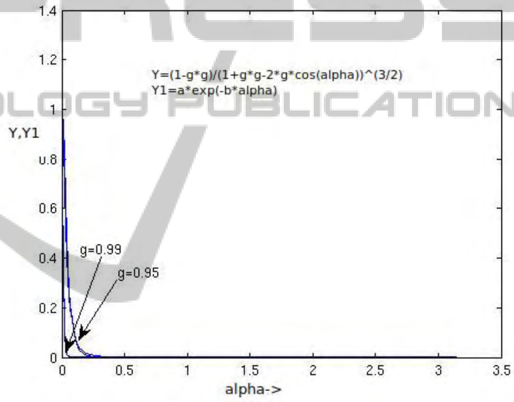

Figure 8: Comparing our approximation to the Henyeye Greenstein approximation for increasing values of phase angle α: $0.95 \leq g \leq 0.99$.

3.2 Backward Scattering Approximation

Although the backward scattering component is usually much smaller than the forward scattering component, we include this as well for completeness. Unfortunately, however, a simple low-error approximation to the kernel for L_2 seems harder to find. Instead we are able to better approximate the ratio between the forward and the backward parts of the integral. Observing the nature of the resulting ratio, we approximate it with the following exponential:

$$\frac{\int_0^{\theta_1} e^{-\kappa(\sec\theta+\tan\theta)}d\theta}{\int_0^{\frac{\pi}{2}} e^{-\kappa(\sec\theta-\tan\theta)}d\theta} = f(\theta_1)e^{-g(\theta_1)\kappa^{h(\theta_1)}}$$

where,

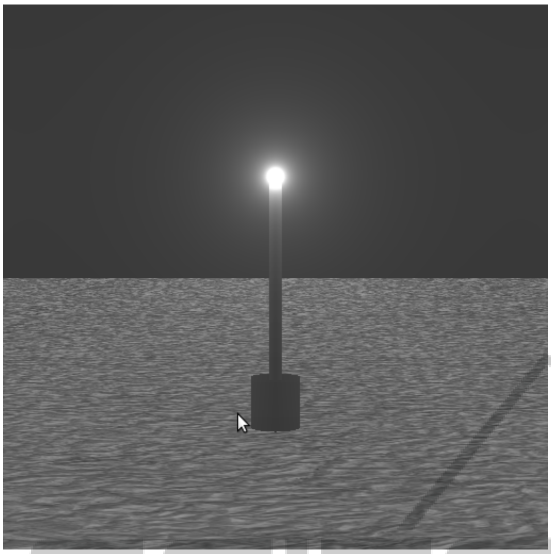

Figure 9: Rendered airlight using Henyeye Greenstein phase function with g=0.7 and $\beta = 0.05$.

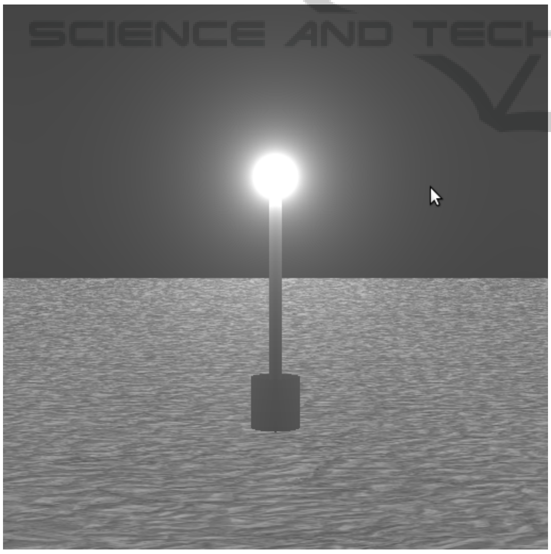

Figure 10: Rendered airlight using Henyeye Greenstein phase function with g=0.4 and $\beta = 0.05$.

$$\begin{bmatrix} f(\theta_1) \\ g(\theta_1) \\ h(\theta_1) \end{bmatrix} = \begin{bmatrix} -0.0394 & 0.1065 & 0.5774 & 0.0058 \\ -0.1299 & 0.4204 & 0.3811 & 0.5993 \\ 0.0171 & -0.1586 & -0.0424 & 1.0879 \end{bmatrix} \begin{bmatrix} \theta_1^3 \\ \theta_1^2 \\ \theta_1 \\ 1 \end{bmatrix}$$

Achieving quantitatively small fractional error for backward scattering with Henyey GreenStein phase function remains an open problem. However, since the magnitude of this component is usually smaller, even large fractional error does not produce noticeable differences in the final image. The error graph is shown in Figure 11.

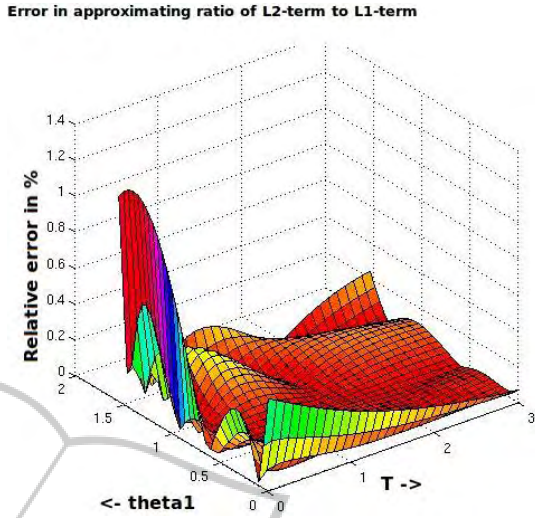

Figure 11: The error in the radiance due to the back-scatter approximation by relating the integral to the forward-scattering.

4 EXPERIMENTS AND RESULTS

We have implemented our approximations on an Intel Centrino based linux laptop PC with 4GB RAM and an nVIDIA GT-240M graphics processor. We implemented the integration in the pixel shader and are able to render scenes such as shown in Figure 18 more than 90 times a second on a 512x512 pixel screen. The illumination behavior is as expected. As the value of the extinction coefficient β becomes smaller, the glow size decreases (Figures 12 and 13). Also, the glow sizes of distant light is smaller than the one closer (Figure 14).

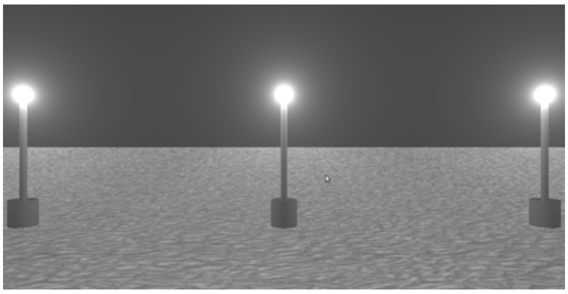

Figure 12: Rendered using pixel shader. $\beta = 0.01$. Lower β implies smaller glows.

Figures 15 shows an example demonstrating that backward scattering has a small effect on the images.

We compare our rendering of the airlight to a povRay based implementation of Monte Carlo integration in Figure 16. We use the intensity distribution of the glow to demnstrate the difference because full

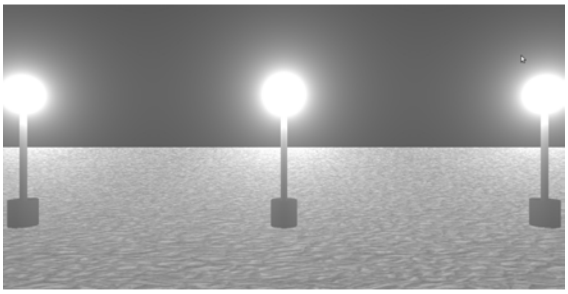

Figure 13: Rendered using pixel shader. $\beta = 0.04$. As β increases, glow does too.

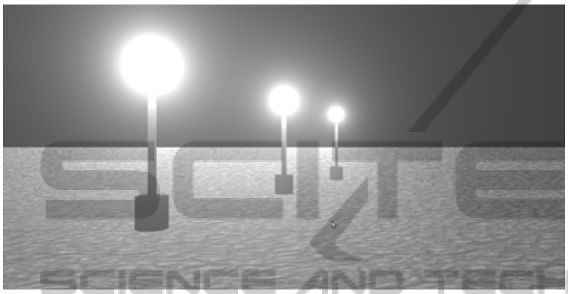

Figure 14: Rendered using pixel shader. $\beta = 0.04$. Far away light has less glow.

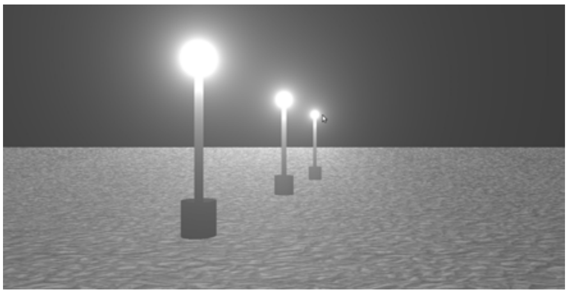

Figure 15: Rendered using pixel shader. Forward scattering only $\beta = 0.04$.

scene renderings tend to hide differences. Although the differences are noticeable, the overall qualitative structures of the scenes are comparable. As a point of reference, the povRay implementation takes 10 seconds to render this scene while our pixel shader based implementation renders the same scene more than a thousand times a second.

5 CONCLUSIONS

Out goal was to find an accurate and efficient approximation for the single scattering integral. As a result of this, the entire integral can be re-evaluated at each pixel at interactive rates. Since there is no precomputation, changes to the particle density, for example due to increasing rain, can be easily handled.

We have not focussed on surface scattering effects or shadows, which can easily be incorporated following (Sun and Ramamoorthi, 2005) and (Biri et al., 2006). For example, we demonstrate surface scattering effect in Figures 17 and 18. The errors introduced by our approximations are quite low and we feel the derived coefficient would be useful to many others. At the same time, even with an unoptimized code rendering is fast.

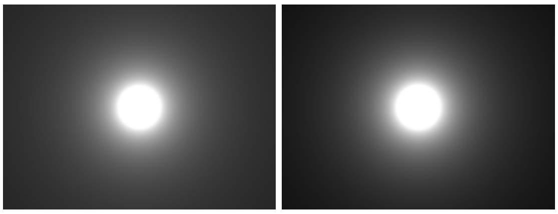

Figure 16: Comparison of our rendering (left) with povRay's (using istropic phase function).

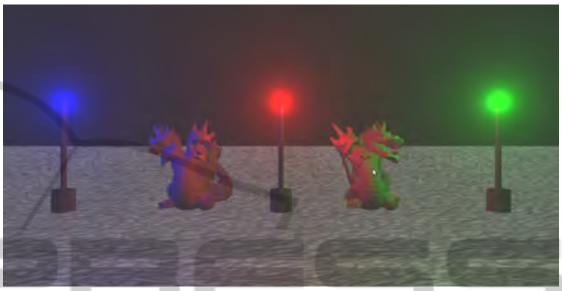

Figure 17: A scene with a dragon model rendered without surface scattering.

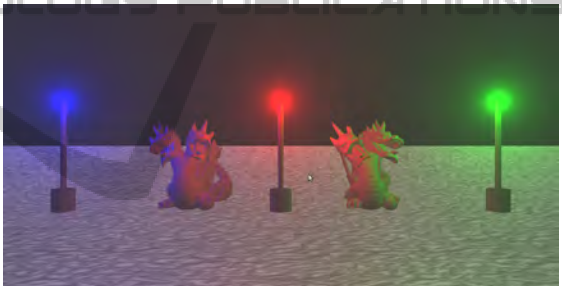

Figure 18: A scene with a dragon model rendered with surface scattering. The dragons consist of more than 100,000 triangles.

REFERENCES

Arbree, A., Walter, B., and Bala, K. (2010a). Heterogeneous subsurface scattering using the finite element method. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints).

Arbree, A., Walter, B., and Bala, K. (2010b). Heterogeneous subsurface scattering using the finite element method. *IEEE Transactions on Visualization and Computer Graphics*, 99(PrePrints).

Bernabei, D., Ganovelli, F., Pietroni, N., Cignoni, P., Pattanaik, S., and Scopigno, R. (2010). Real-time single scattering inside inhomogeneous materials. *Vis. Comput.*, 26(6-8):583–593.

Biri, V., Michelin, S., and Arques, D. (2006). Real time rendering of atmospheric scattering and volumetric shadows. In *Journal of WSGC*, pages 65–72.

Blinn, J. F. (1982). Light reflection functions for simulation of clouds and dusty surfaces. In *Proceedings of the 9th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '82, pages 21–29, New York, NY, USA. ACM.

Cerezo, E., Perez, F., Pueyo, X., Seron, F. J., and Sillion, F. X. (2005). A survey on participating media rendering techniques. *The Visual Computer*, 21(5):303–328.

Chandrasekhar, S. (1960). *Radiative Transfer*. Dover Publications.

Henyey, L. and Greenstein, J. (1941). Diffuse radiation in the galaxy. *Astrophys. Journal*, 93:70–83.

Jensen, H. W. and Christensen, P. H. (1998). Efficient simulation of light transport in scences with participating media using photon maps. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '98, pages 311–320, New York, NY, USA. ACM.

Lafortune, E. (1996). *Mathematical Models and Monte Carlo Algorithms for Physically Based Rendering*. PhD thesis, Cornell University.

Max, N. L. (1986). Atmospheric illumination and shadows. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '86, pages 117–124, New York, NY, USA. ACM.

Nishita, T., Miyawaki, Y., and Nakamae, E. (1987). A shading model for atmospheric scattering considering luminous intensity distribution of light sources. *SIGGRAPH Comput. Graph.*, 21:303–310.

Premoze, S., Ashikhmin, M., Ramamoorthi, R., and Nayar, S. K. (2004). Practical rendering of multiple scattering effects in participating media. In *Rendering Techniques*, pages 363–373.

Rushmeier, H. E. and Torrance, K. E. (1987). The zonal method for calculating light intensities in the presence of a participating medium. *SIGGRAPH Comput. Graph.*, 21:293–302.

Sillion, F. (1994). Clustering and volume scattering for hierarchical radiosity calculations. In *In Fifth Eurographics Workshop on Rendering*, pages 105–117.

Stam, J. (1994). Stochastic rendering of density fields. In *Proceedings of Graphics Interface 94*, pages 51–58.

Sun, B. and Ramamoorthi, R. (2005). A practical analytic single scattering model for real time rendering. *ACM Trans. Graph*, 24:1040–1049.

Zhou, K., Hou, Q., Gong, M., Snyder, J., Guo, B., and Shum, H.-Y. (2007). Fogshop: Real-time design and rendering of inhomogeneous, single-scattering media. In *Proceedings of the 15th Pacific Conference on Computer Graphics and Applications*, pages 116–125, Washington, DC, USA. IEEE Computer Society.

APPENDIX

Phase Function Approximation

The matrix below are in the format:

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{bmatrix}$$

$$\text{If } 0 \leq g \leq 0.2: \begin{bmatrix} -0.8470 & 0.6117 & 1.0004 \\ 0.7218 & 2.2603 & 0.0009 \end{bmatrix}$$

$$\text{If } 0.2 \leq g \leq 0.4: \begin{bmatrix} -0.3199 & 0.4128 & 1.0197 \\ 3.1063 & 1.2475 & 0.1109 \end{bmatrix}$$

$$\text{If } 0.4 \leq g \leq 0.6: \begin{bmatrix} -0.2697 & 0.3865 & 1.0223 \\ 9.4069 & -4.0092 & 1.2134 \end{bmatrix}$$

$$\text{If } 0.6 \leq g \leq 0.8: \begin{bmatrix} -0.3067 & 0.4298 & 1.0096 \\ 46.9285 & -51.3639 & 16.1729 \end{bmatrix}$$

$$\text{If } 0.8 \leq g \leq 0.9: \begin{bmatrix} -1.1784 & 1.8440 & 0.4358 \\ 369.6793 & -573.3485 & 227.3450 \end{bmatrix}$$

$$\text{If } 0.9 \leq g \leq 0.95: \begin{bmatrix} -7.5 & 13.3 & -4.7 \\ 2829.9 & -5022.1 & 2238.6 \end{bmatrix}$$

$$\text{If } 0.95 \leq g \leq 0.99: \begin{bmatrix} -50 & 90 & -40 \\ 78630 & -150550 & 72090 \end{bmatrix}$$